IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 November 2022, accepted 13 December 2022, date of publication 15 December 2022,
date of current version 8 February 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3229890

RESEARCH ARTICLE

A Novel Method for Few-Shot Specific Emitter Identification in Non-Cooperative Scenarios

CUNXIANG XIE[1], **LIMIN ZHANG**[1], **AND ZHAOGEN ZHONG**[2]
[1]Department of Information Fusion, Naval Aviation University, Yantai 264001, China
[2]The School of Aviation Basis, Naval Aviation University, Yantai 264001, China

Corresponding author: Zhaogen Zhong (zhongzhaogen@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 91538201, in part by the Taishan Scholar Project of Shandong Province under Grant ts201511020, and in part by the Project through the Chinese National Key Laboratory of Science and Technology on Information System Security under Grant 6142111190404.

ABSTRACT Obtaining larger category-label-containing training signal datasets in non-cooperative scenarios is difficult. Moreover, employing smaller labeled signal datasets for specific emitter identification is technically challenging. Therefore, we propose a novel method for few-shot SEI. We first design a bispectral analysis and Radon transformation-based signal preprocessing scheme to obtain feature vectors that effectively characterize the radio frequency fingerprints. The feature vectors are then fed to a network model for feature learning. Moreover, a meta-learning algorithm is applied to the network model to adapt to few-shot feature learning. The conventional meta-learning algorithm is improved to develop a novel algorithm involving latent embedding optimization for meta-learning. The proposed method extracts low-dimensional key features from high-dimensional input data and evaluates the distance and degree of feature dispersion. The resulting information is employed in sample point prediction. The algorithm effectively achieves few-shot SEI and offers stable and efficient recognition after training with a minimum of forty samples. This method identifies emitter individuals under multiple modulation types and exhibits scalability in identifying the emitter numbers. Moreover, it offers adaptability in identifying the emitter individuals under multiple propagation channel types.

INDEX TERMS Few-shot, latent embedding optimization, meta-learning, non-cooperative communication, specific emitter identification.

I. INTRODUCTION

Specific emitter identification (SEI) [1], [2] is a technique for identifying specific emitters by extracting the subtle features contained in the radio frequency (RF) signals emitted by the emitters. Different manufacturers use different manufacturing techniques for different individual emitters, even for the same model of emitter equipment, which leads to minor differences between individual hardware architecture. Hardware differences are inevitably introduced during subsequent equipment assembly and testing. Hardware differences are expressed by RF signals that exhibit a stable and regular waveform characteristic of signals and uniquely identify the individual emitter to which they belong; hence, they are also called RF fingerprints (RFFs) [3], [4]. Different individual emitters can be identified by mining and extracting the RFFs of the signal. The technology is independent of the communication method and content but directly extracts the RFFs of the physical layer to identify individual emitters. In terms of application, with the development of radio technology in various industries, effective management of the radio spectrum and monitoring of the infringement of abnormal spectrum has become prominent. The government can implement effective radio management through the SEI technology, directly distinguishing legal and illegal radio users through the physical level of RF signals in the complex radio spectrum, and monitoring and tracking the individual radio stations corresponding to the harmful spectrum [5], [6], [7].

The key to the SEI technology lies in the extraction of features of RFFs and the subsequent classification and identification of individuals according to the feature difference. Recently, extensive research on SEI has been conducted.

The associate editor coordinating the review of this manuscript and approving it for publication was Meng-Lin Ku.

Padilla et al. [8] proposed a method of extracting RFFs using the spectrum information of communication preamble and successfully identified several Wi-Fi devices. Owing to undesirable hardware differences, the in-phase (I) and quadrature (Q) components of the signal introduce amplitude and phase deviations during modulation, called IQ imbalance deviations. Klein et al. [9], [10] proposed a method for SEI based on the dual-tree complex wavelet transform that decomposes the signal using the dual-tree complex wavelet transform and extracts the wavelet coefficients as the characteristic parameters for individual identification. Yuan et al. [11] performed Hilbert-Huang transformations on transient RF signals generated by eight individual emitters to obtain the Hilbert time-frequency energy distribution, based on which 13 time-frequency characteristic parameters were designed and extracted as RFFs to accomplish the task of SEI. Zhang et al. [12] used the Hilbert-Huang transformation to obtain the Hilbert time-frequency energy spectrum of the signal and subsequently proposed three different feature extraction methods: 1. Calculation of the entropy of the Hilbert spectrum and the first and second order moments of the signal as the identification features. 2. Calculation of the correlation coefficient of the Hilbert spectrum as the identification features. 3. Extraction of the elements of the Hilbert spectrum with high discrimination as the identification features using the Fisher discriminant method. The study in [12] compared the advantages and disadvantages of three methods and discussed the identification performance of the algorithms in single-hop and relaying scenarios. Satija et al. [13] decomposed RF signals by variational mode decomposition (VMD) in the time and frequency domain modes and subsequently extracted the VMD entropy and cumulative value as the RFFs to effectively achieve SEI in the single-hop and relaying scenarios.

The complexity of RFFs cannot be represented by a unified mathematical model. The characteristic parameters extracted by conventional methods cannot fully characterize the features of emitters, and the identification performance improvement is limited. With the development of artificial intelligence technology [14], [15], [16], [17], [18], deep learning has been applied to SEI applications. This novel research direction can comprehensively extract the features of emitter signals through neural networks to improve the identification performance. Merchant et al. [2] achieved 92.29% identification accuracy for seven ZigBee devices based on the feature extraction of the original time-domain baseband signal using convolutional neural networks (CNNs). Ding et al. [19] first calculated the bispectrum distribution of the steady-state RF signal, subsequently used a supervised dimensionality reduction method [20] to compress the bispectrum to significantly reduce the computational complexity, and finally used a CNN to extract and identify the features of the compressed bispectrum. Based on the experimental results, the method can effectively identify five individual emitters with 90% accuracy at a signal-to-noise ratio (SNR) of 12 dB, which proves the effectiveness of the method. Pan et al. [21] performed the Hilbert-Huang transformation on the received RF signal to obtain the Hilbert spectrum and converted it to grayscale images. They subsequently constructed a deep residual network to learn the visual differences reflected by the Hilbert grayscale images to achieve the differentiation of different types of signals. This method can accommodate SEI under various channels, such as additive white Gaussian noise (AWGN), Rayleigh fading, and frequency-selective channels. Wong et al. [22] used the I/Q components of RF communication signals as inputs to a CNN to estimate the amplitude and phase deviations of the signals as characteristic parameters for the classification. However, this method should be performed based on the same modulation method and is less expandable. He et al. [23] performed signal decomposition on the received signal to obtain the non-Gaussian characteristics of the signal and subsequently used a long short-term memory (LSTM) network to extract deep features. This method does not require a priori knowledge of the signal, and LSTM is more suitable for processing one-dimensional periodic data such as signals than CNNs; thus, it has the advantage of high identification rate and low complexity. Wang et al. [24] proposed an efficient SEI method based on a complex-valued neural network and network compression. The former is used to directly process complex baseband signals to extract their RFFs; the latter is used to reduce the model size and computational complexity while ensuring the efficient performance of the network model. Based on the experimental results, this method has superior identification accuracy and network convergence performance compared with the existing deep learning-based SEI methods, and the proposed compression algorithm can reduce the network size to 10–30% of the original size while ensuring almost constant identification performance. Zha et al. [25] proposed a complex Fourier neural operator and embedded it in a neural network to form a complex Fourier neural network (CFNN) that can learn the RFFs of a signal from the time and frequency domains.

SEI based on deep learning is a recent research direction. Despite its numerous prospects, deep learning is a data-driven approach that requires numerous training data to realize appropriate performance. However, in practical application scenarios, particularly in non-cooperative communication scenarios, obtaining a large number of signal samples containing tag information is difficult. Performing the SEI under the condition of "few-shot" is a problem that needs to be focused on. Qian et al. [26] proposed a multi-level sparse representation-based identification (MSRI) algorithm for few-shot SEI. MSRI uses the channel attention mechanism to concatenate the shallow and deep RF signals extracted by the CNN and subsequently constructs a multi-layer dictionary for sparse representation of RFFs to achieve efficient SEI under few-shot conditions. The method uses at least 15 training samples to identify nine independent communication emitters with an identification accuracy that exceeds 90%. Yang et al. [27] used meta-learning to address few-shot SEI. This method incorporates the meta-learning mechanism into a deep learning model with a simple algorithm and can achieve an identification accuracy above 90% using only 20 training samples.

This study proposes a novel method to solve few-shot SEI. First, a signal preprocessing scheme based on bispectral analysis and Radon transform is designed to obtain the characteristic vectors that can effectively characterize the RFFs of signals, thus considerably avoiding the problems of in-depth feature mining and incomplete feature extraction caused by the direct processing of RF signals by neural network models. Then the characteristic vectors are subsequently fed into the neural network for feature learning, and a meta-learning mechanism is introduced to enable the network to achieve feature learning under few-shot conditions. Meanwhile, the use of conventional meta-learning algorithms in the direct processing of high-dimensional input data results in poor model training results and low generalization owing to the complexity of RFFs and the subtlety of the differences between different types of RFFs. Finally, this study proposes a latent embedding optimization (LEO) for meta-learning that extracts key features with low dimensionality from high-dimensional input data, calculates the distance and scatter between these features, and uses this information to predict sample points. The LEO algorithm considers the similarity among sample points and the scatter of sample points, integrates the distribution characteristics for each type of sample point, and normalizes the distances that cause the distances between different types of sample points to be comparable and more accurate in classification. Therefore, the proposed method can effectively resolve the few-shot SEI problem, and the identification performance is better than state-of-the-art results.

The remainder of this paper is organized as follows. In section II, we introduce the signal pre-processing scheme based on bispectral analysis and radon transform; In section III, we introduce the proposed few-shot SEI method based on meta-learning with latent embedding optimization; In section IV, we present and discuss the experimental results. Finally, we conclude the paper.

II. SIGNAL PREPROCESSING

A. BISPECTRAL ANALYSIS

Bispectral analysis, as a higher-order spectrum analysis with the lowest order, shows superiority in addressing non-Gaussian and non-stationary signals. The bispectral analysis of a signal is essentially a two-dimensional Fourier transform of the third-order cumulative quantity of the signal that can be expressed as follows [28]:

$$B(\omega_1, \omega_2) = \sum_{\tau_1} \sum_{\tau_2} C_{3x}(\tau_1, \tau_2)\, e^{-j(\omega_1\tau_1 + \omega_2\tau_2)} \tag{1}$$

where ω_1 and ω_2 denote the two-dimensional Fourier transform frequency, and $C_{3x}(\tau_1, \tau_2)$ denotes the third-order cumulative quantity that can be expressed as follows:

$$C_{3x}(\tau_1, \tau_2) = E\left[x(t)\, x(t+\tau_1)\, x(t+\tau_2)\right] \tag{2}$$

B. RADON TRANSFORM

The bispectrum distribution can suitably characterize the RFFs of the signal of the communication emitter; however, the computation complexity increases and processing efficiency decreases by directly using the two-dimensional bispectrum distribution to perform the subsequent processing. Therefore, the Radon transform is introduced to project the two-dimensional bispectrum distribution onto a one-dimensional characteristic vector. The Radon transform [29], [30] calculates the linear integral of a two-dimensional function over a two-dimensional plane xOy with respect to any line. The Radon transform of a bispectrum distribution $B(\omega_1, \omega_2)$ can be expressed as follows:

$$R(\alpha, \rho) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} B(\omega_1, \omega_2)\, \delta(\rho - \omega_1 \cos\alpha - \omega_2 \sin\alpha) \times d\omega_1 d\omega_2 \tag{3}$$

where α denotes the angle between the line and the coordinate axis, and ρ denotes the perpendicular distance between the line and the origin. The characteristic function δ is defined as

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases} \tag{4}$$

to ensure that the integration is performed along the line $\rho = \omega_1 \cos\alpha + \omega_2 \sin\alpha$. Here, the Radon transformation is performed by considering $\rho = 0$ and $\alpha = 3\pi/4$; that is, linear integration with respect to line $\omega_1 = \omega_2$, and the result is normalized.

The bispectrum distribution of the two types of RF signals $x_1(t)$, $x_2(t)$ with their Radon projection characteristic maps, is shown in Fig. 1, according to which a significant difference in the bispectrum distributions of signals $x_1(t)$ and $x_2(t)$ can be observed. Meanwhile, the two Radon projection characteristic maps demonstrate subtle but stable differences, proving the effectiveness of the Radon projection characteristic vector in characterizing RFFs, distinguishing different emitters, and performing SEI.

III. IMPROVED META-LEARNING ALGORITHM

A. META-LEARNING

In contrast to conventional machine learning, meta-learning [31] considers improving the problem-solving ability of meta-learning models by training some tasks such that they can provide suitable solutions to problems that have not been solved previously. Meta-learning is performed by the cooperation of the base and the meta-learner. The functions of the base learner and meta-learner are to learn the characteristics of each task and the commonalities of all tasks, respectively, both of which are independent on each other, but interact with each other. The base learner delivers the results of each training task and the key training parameters that need to be stored in the memory module to the meta-learner. In the framework of the combined base learner and meta-learner, the learning of task characteristics is maximized in the base learner, and the learning of task commonalities is maximized

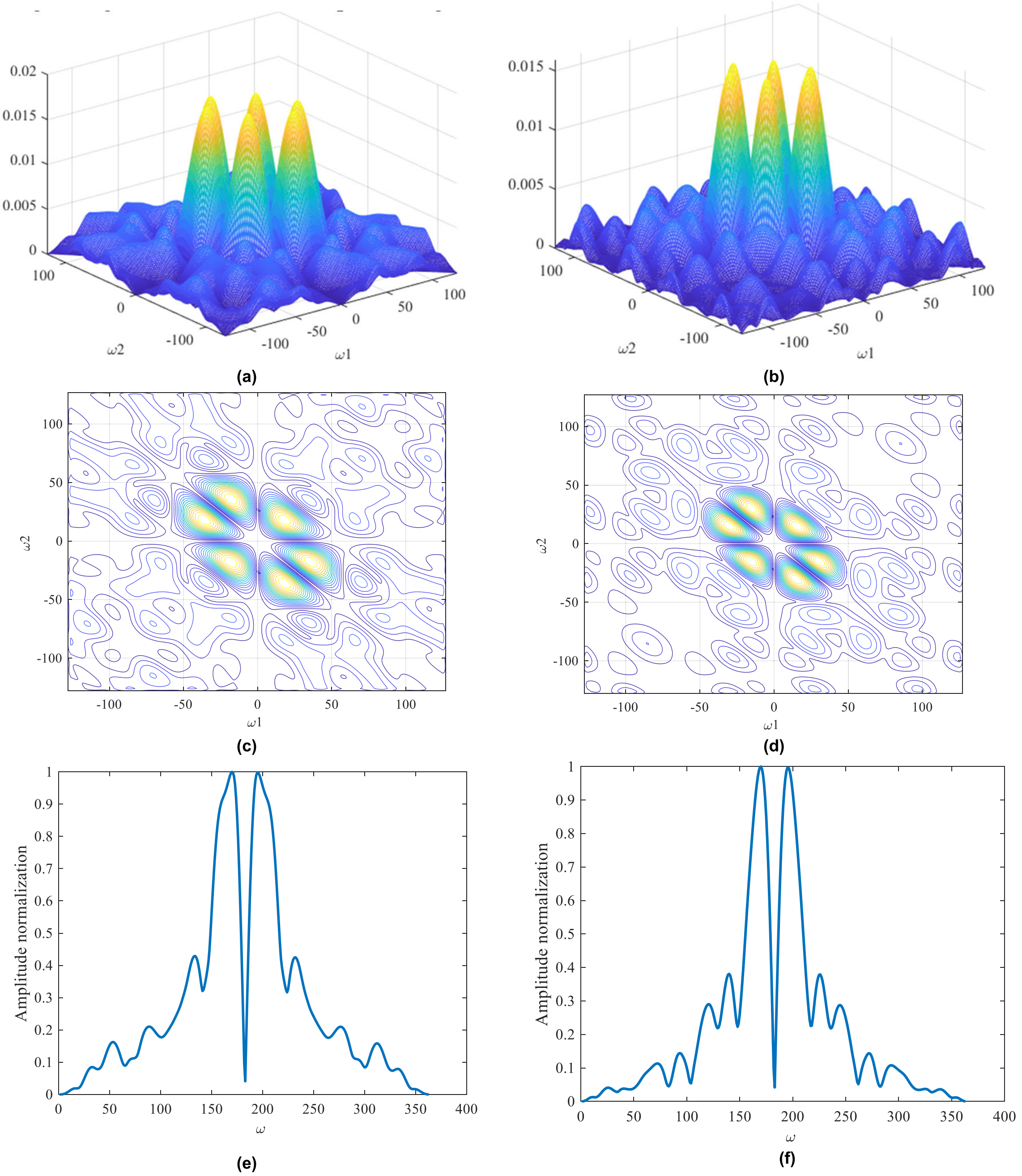


FIGURE 1. Bispectral distribution and Radon projection features: (a) three-dimensional bispectral distribution of $x_1(t)$; (b) three-dimensional bispectral distribution of $x_2(t)$; (c) two-dimensional bispectral distribution of $x_1(t)$; (d) two-dimensional bispectral distribution of $x_2(t)$; (e) Radon projection feature of $x_1(t)$; (f) Radon projection feature of $x_2(t)$.

in the meta-learner, thus allowing the meta-learning framework to achieve high accuracy on both the training and test datasets. The procedures of the meta-learning algorithm are represented as follows:

First, the base learner on the task is denoted as f_θ, where θ is the trainable parameter in the base learner. In fragmented training, we extracted the ith task T_i that features randomness and obeys the task distribution. For task T_i, the loss function of the base learner is $L_{T_i}(f_\theta)$. In the training of the base learner, the objective function to be optimized is $\min_\theta L_{T_i}(f_\theta)$.

Second, ϕ is the initial value of the parameters provided by the meta-learner to the base learner, and an iteration is performed on task T_i to update the trainable parameters.

$$\phi_i^{(1)} = \phi - \alpha \left[\nabla_\theta L_{T_i}(f_\theta)\right]_{\theta=\phi} \tag{5}$$

where $\left[\nabla_\theta L_{T_i}(f_\theta)\right]_{\theta=\phi}$ is the value of the gradient of the loss function of task T_i at the initial value ϕ of the parameter, α is the step parameter, and $\phi_i^{(1)}$ is the parameter obtained after one iteration of updating the initial value of the parameter. The number of iterations may be more than one and depends on the number of samples in the task dataset. The larger the amount of labeled data available, the higher the number of iterations allowed, and improving the accuracy of the base learner for the task is more beneficial. For example, in the second iteration, the trainable parameters are updated as follows:

$$\phi_i^{(2)} = \phi_i^{(1)} - \alpha \left[\nabla_\theta L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(1)}} \tag{6}$$

where $\left[\nabla_\theta L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(1)}}$ is the value of the gradient of the loss function of task T_i considered at the parameter value $\phi_i^{(1)}$ after one iteration, and $\phi_i^{(2)}$ is the parameter obtained after the initial value of the parameter has been updated in two iterations. After the Nth iteration, the trainable parameters are updated as follows:

$$\phi_i^{(N)} = \phi_i^{(N-1)} - \alpha \left[\nabla_\theta L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N-1)}} \tag{7}$$

where $\left[\nabla_\theta L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N-1)}}$ is the value of the gradient of the loss function of task T_i considered at parameter value $\phi_i^{(N-1)}$ after $N-1$ iterations, and $\phi_i^{(N)}$ is the parameter obtained after N iterations of updating the initial value of the parameter.

Third, the gradient values $\left[\nabla_\theta L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N)}}$ of the loss function after N iterations of the task are fed back to the meta-learner that subsequently uses these gradients to update the initial values of the parameters in the memory module, and the meta-objective function is expressed as follows:

$$\min_{\phi} \sum_{T_i \sim p(T)} L_{T_i}\left(f_{\phi_i^{(N)}}\right) \tag{8}$$

where $L_{T_i}\left(f_{\phi_i^{(N)}}\right)$ is the value of the loss function of task T_i at parameter value $\phi_i^{(N)}$. The parameter value $\phi_i^{(N)}$ is a function of the initial value of the parameter ϕ, the step parameter α, and the task data set. The meta-objective function is the summation of the loss function values of the validation set on all tasks; subsequently, the initial value of the parameter ϕ is searched to minimize the meta-objective function, that is, the minimization of the summation of the loss function on all tasks. The initial values of the parameters computed by optimizing the meta-objective function allow the minimization of the loss function on the task; thus, they are closer to the optimal values of the parameters in the base learner on the task that requires only a fewer iterative update rounds to obtain improved parameter estimates, allowing the base learner to achieve higher accuracy on the task.

Fourth, the meta-learner uses the loss gradient values to update the initial values of the parameters in the memory module.

$$\phi \leftarrow \phi - \beta \sum_{T_i \sim p(T)} \nabla_\phi \left[L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N)}} \tag{9}$$

where ϕ is the initial value of the parameter in the memory module, β is the step parameter used in the meta-learner when optimizing the meta-objective function, $\left[L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N)}}$ is the value of the loss function on task T_i considered at parameter value $\phi_i^{(N)}$, and $\nabla_\phi\left[L_{T_i}\left(f_\theta\right)\right]_{\theta=\phi_i^{(N)}}$ is the gradient of the loss function value on the task after iterative training with respect to the initial value of the parameter. This gradient is used to update the initial value of the parameter in the meta-learner, such that the initial value of the parameter is updated in the direction that will eventually reduce the loss function value of the task. The initial values of the parameters in the meta-learner can be updated using the formula of stochastic gradient descent, and when a new task is later encountered, the meta-learner provides this updated initial value of the parameters to the base learner of the new task.

With a limited number of training samples, only few iterations can be performed to update the parameters in the base learner. Therefore, the parameters in the base learner must have suitable initial values from the meta-learner. The meta-learner combines the training experience of all tasks and feeds the gradient of the loss function and the initial values of the model to the base learner, allowing the base learner to achieve suitable accuracy after fewer iterations. The meta-learner and base learner efficiently function together, with the base learner in charge of learning the characteristics of each task, and the meta-learner integrates the experience of the base learner and provides the information required by the base learner to accelerate its training and improve the accuracy of the training results.

B. LATENT EMBEDDING OPTIMIZATION

Based on previous analysis, the meta-learning algorithm exhibits unique effectiveness and superiority in handling few-shot problems. However, regarding few-shot SEI, the use of conventional meta-learning algorithms in the direct processing of high-dimensional input data results in a poor model training effect and low generalization ability owing to the complexity of RFFs and the subtlety of the differences between different types of RFFs. Therefore, this section proposes an improved meta-learning algorithm, termed the LEO algorithm that extracts key features with low dimensionality from high-dimensional input data and calculates the distance and scatter between these features and uses this information to predict sample points. The LEO algorithm considers the similarity between all sample points and the scatter of sample points, integrates the distribution characteristics of each type of sample point, and normalizes the distances that cause the distances between different types of sample points to be similar and the classification to be more accurate. In the meta-learner, the loss function of the task validation set is minimized using stochastic gradient descent (SGD) that maximizes the generalization ability of the model, and the meta-parameters are computed. The meta-learner feeds the meta-parameters to the base learner. Subsequently, the base learner minimizes the loss function of the task training set to provide prediction results for the task.

The LEO algorithm includes a base learner and a meta-learner, and also includes an encoder and a decoder. In the base learner, the encoder maps the high-dimensional input data into characteristic vectors. The decoder maps the characteristic vectors of the input data into the probability values of the input data falling into each category. The base learner performs one iteration of parameter updates using the meta-parameters provided by the meta-learner to provide the prediction results of data labeling. The meta-learner provides meta-parameters for the encoder and decoder of the base learner. These meta-parameters include the parameters of the feature extraction model, encoder, and decoder. They are updated by minimizing the generalization error for all the tasks, that is, minimizing the summation of the loss functions of all the task validation sets. The base learner of the LEO algorithm is highly efficient, whereas the computation in the meta-learner is relatively slow and complex. The base learner can speedily accommodate new tasks and provide accurate results, whereas the meta-learner is relatively slow in integrating the characteristics of all tasks and updating the meta-parameters. All the complex parameter updates are performed by the meta-learner, and the base learner performs only efficient computations, thus allowing adaptation to the task characteristics. This increases the training efficiency of the entire meta-learning framework and can process similar tasks that can rapidly reach high accuracy.

The base learner includes complex structures, that is, encoders and decoders, that are used to calculate the probability at which the input data falls under each category. However, the parameters of the base learner are updated in the meta-learner. As the base learner does not have to compute these parameters, it simply considers the meta-parameters provided by the meta-learner, performs efficient model computation, updates the parameters for each iteration, provides a predictive labeling on the task, and achieves a high classification accuracy on the task. First, the model structure of the base learner including the encoder and decoder is introduced here. The encoder model consists of two main components:

(1) Encoder g_{ϕ_e}, where ϕ_e is a trainable parameter of the encoder that is a neural network model used to extract features from the input data either using a CNN or a residual neural network model.

(2) The correlation network g_{ϕ_r}, where ϕ_r is the trainable parameter of the correlation network that is a depth metric model used to calculate the distance between the features and can employ either a CNN or a fully connected neural network model.

The features of the input data of the nth category are denoted as z_n. For the input data, the feature extraction process is first performed on the input data belonging to the nth category using the encoder g_{ϕ_e}; subsequently, the distances between features are calculated using the correlation network g_{ϕ_r}. The mean and scatter of these distances are calculated considering the distances between all sample points in the training set; the features z_n of the input data of the nth category follow the Gaussian distribution. The expectation and variance of the Gaussian distribution are the mean and scatter of these distances, respectively. The specific calculation formula is as follows:

$$\mu_n^e,\ \sigma_n^e = \frac{1}{NK^2}\sum_{k_n=1}^{K}\sum_{m=1}^{N}\sum_{k_m=1}^{K} g_{\phi_r}\left[g_{\phi_e}\left(x_n^{k_n}\right), g_{\phi_e}\left(x_m^{k_m}\right)\right] \tag{10}$$

$$\boldsymbol{z}_n \sim q\left(\boldsymbol{z}_n|\boldsymbol{D}_n^{tr}\right) = N\left\{\mu_n^e, diag\left(\sigma_n^e\right)^2\right\} \tag{11}$$

where N is the number of categories, K is the number of samples in each category, and $\boldsymbol{D}_n^{tr}$ is the training dataset for the nth category. Each category of input data has$K$ samples, and the distance between these K samples and all known images is calculated. Two models are included in the distance calculation: the feature extraction model and the depth metric model. The mean and variance of these distances are calculated. The feature $\boldsymbol{z}_n$ of the input data of the nth category follows a Gaussian distribution, and the expectation and variance of the Gaussian distribution are the mean and variance of these distances, respectively. There are N categories in total, and the potential features of all the categories denoted as $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n, \cdots \boldsymbol{z}_N)$ are determined based on the calculation of the encoder.

The decoder model maps the feature vector $\boldsymbol{z}_n$ of the input data for each category to the probability values $\boldsymbol{w}_n$ falling under each category.

$$\mu_n^d,\ \sigma_n^d = g_{\phi_d}\left(\boldsymbol{z}_n\right) \tag{12}$$

$$\boldsymbol{w}_n \sim q\left(\boldsymbol{w}|\boldsymbol{z}_n\right) = N\left\{\mu_n^d, diag\left(\sigma_n^d\right)^2\right\} \tag{13}$$

where the base learner parameter of task T is denoted as $\boldsymbol{w}_T$. The base learner parameter consists of the probability values that the input data fall under each category denoted as $\boldsymbol{w}_T = (\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_N)$. The base learner parameter $\boldsymbol{w}_n$ refers to the probability value of the input data falling under the nth category, g_{ϕ_d} is the mapping from the feature vector to the base learner parameter, and ϕ_d is the trainable parameter in the decoder.

The function of the encoder is to map the input data of the nth category to the feature vector of the nth category, and the function of the decoder is to map the feature vector of the nth category to the probability value that the input data falls under the nth category. The encoder and decoder are both in the base learner and are used to calculate the probability values of the input data falling under each category, and thus, they classify the input data. The meta-learner provides the parameters in the encoder and decoder, and the base learner efficiently performs the classification of the input data with the encoder and decoder to achieve a high classification accuracy on the task. After task training is completed, the base learner inputs the feature vector of data from each category and the base learner parameters $\boldsymbol{w}_T$ for task T into the meta-learner that uses this information to update the meta-parameters.

In the base learner, the loss function for task T is the cross-entropy loss function that measures the difference between the probability of the input data falling under its true category and the probability of falling under other categories. Optimizing the cross-entropy loss function allows the maximization of the probability that the input data falls under its true category and the minimization of the probability that the input data falls under other categories. The loss function of the training set for task T can be expressed as follows:

$$L_T^{tr}\left(f_{w_T}\right) = \sum_{(\boldsymbol{x},y)\in \boldsymbol{D}_T^{tr}} \left[-\boldsymbol{w}_y\boldsymbol{x} + \ln\sum_{j=1}^{N} e^{\boldsymbol{w}_j\boldsymbol{x}}\right] \tag{14}$$

where $(\boldsymbol{x}, y)$ are the sample points in training set $\boldsymbol{D}_T^{tr}$ for task $T, f_{\boldsymbol{w}_T}$ is the base learner for task T. The loss function of task T is minimized to update the exclusive parameter $\boldsymbol{w}_T$ of the task. In the decoder model, the exclusive parameter of the task is $\boldsymbol{w}_n \sim q\left(\boldsymbol{w}|z_n\right)$. Updating the task exclusive parameter $\boldsymbol{w}_T$ implies the update of the feature vector z_n. Here, one round of task exclusive parameter update is performed using SGD.

$$\boldsymbol{z'}_n = z_n - \alpha\nabla_{z_n} L_T^{tr}\left(f_{\boldsymbol{w}_T}\right) \tag{15}$$

where $\boldsymbol{z'}_n$ is the updated feature vector, corresponding to the updated task exclusive parameter $\boldsymbol{w'}_T$. α is the learning rate parameter of the base learner that uses $\boldsymbol{w'}_T$ to predict the labeling of the data in the task validation set. The loss function $L_T^{val}\left(f_{\theta'_T}\right)$ of the validation set $\boldsymbol{D}_T^{val}$ of task T, the updated feature vector $\boldsymbol{z'}_n$, and the updated task exclusive parameters $\boldsymbol{w'}_T$ are input into the meta-learner, where the meta-parameters are updated.

The summation of the loss functions of the validation set for all tasks T is minimized in the meta-learner, and the generalization error of the model on the task is minimized.

$$\min_{\phi_e,\phi_r,\phi_d} \sum_T \Big[L_T^{val}\left(f_{\theta'_T}\right) + \beta D_{KL}\left\{q\left(z_n|\boldsymbol{D}_T^{tr}\right)||p\left(z_n\right)\right\} + \gamma\left\|stopgrad\left(z'_n\right) - z_n\right\|_2^2\Big] + R \tag{16}$$

where $L_T^{val}\left(f_{\theta'_T}\right)$ is the loss function of the validation set of task T. It measures the generalization error of the base learner model. The smaller the loss function, the better the generalization ability of the model. $p\left(z_n\right) = N\left(\boldsymbol{0}, \boldsymbol{I}\right)$ is the standard normal distribution and $D_{KL}\left\{q\left(z_n|\boldsymbol{D}_T^{tr}\right)||p\left(z_n\right)\right\}$ is the KL-divergence between the approximate posterior distribution $q\left(z_n|\boldsymbol{D}_T^{tr}\right)$ and the prior distribution $p\left(z_n\right)$. Minimizing the KL-divergence allows the estimation of the posterior distribution $q\left(z_n|\boldsymbol{D}_T^{tr}\right)$ to be considerably accurate. Minimizing the distance $\left\|stopgrad\left(\boldsymbol{z'}_n\right) - z_n\right\|_2^2$ minimizes the distance between the initial value of the parameter z_n and the updated value $\boldsymbol{z'}_n$ of the parameter after the training is completed, such that the initial and the final values of the parameter are closer, and multiple iterations of updates are not required. R is the regularization term, t used in regulating the complexity of the meta-parameters to avoid overfitting. The regularization term R is calculated as follows:

$$R = \lambda_1\left(\|\phi_e\|_2^2 + \|\phi_r\|_2^2 + \|\phi_d\|_2^2\right) + \lambda_2\left\|\boldsymbol{C}_d - \boldsymbol{I}\right\|_2 \tag{17}$$

where $\boldsymbol{C}_d$ is the correlation matrix between the rows of parameter ϕ_d. $\|\boldsymbol{C}_d - \boldsymbol{I}\|_2$ enables $\boldsymbol{C}_d$ to approach the identity matrix $\boldsymbol{I}$, such that a relatively weak correlation exists between the rows of parameter ϕ_d, the feature vectors of each category, and the probability values falling under each category, thus ensuring an accurate classification.

In summary, the LEO algorithm considers the mean and scatter of the distances between all signal samples under each category and subsequently classifies the signal samples that incorporate more information when compared with the conventional meta-learning algorithm, leading to higher classification accuracy. Additionally, the LEO algorithm uses a complex base learner for complex tasks, whereas the complex parameters of the base learner are updated in the meta-learner. Similar to conventional meta-learning algorithms, the base learner in this algorithm uses SGD to update the parameters of one round of tasks to fit the task characteristics that can achieve higher classification accuracy on the task. Conversely, the conventional meta-learning algorithm for SGD optimization solution involves the calculation of higher-order derivatives of the loss function that can result in slow and unstable training of the meta-learning framework. The base learner of the LEO algorithm uses an encoder and decoder structure that considerably avoids such problems and stabilizes the training.

C. ALGORITHM FLOW

The procedures of the LEO algorithm are as follows:

1. Preset the learning rate parameters α and η, initialize the meta-parameters, encoder parameter ϕ_e, correlation network parameter ϕ_r, decoder parameter ϕ_d, and denote the meta-parameters ϕ_e, ϕ_r, ϕ_d as ϕ.
2. Randomly select task $T, \boldsymbol{D}_T^{tr}$ is the training set for task T, and $\boldsymbol{D}_T^{val}$ is the validation set for task T.
3. Encoded the training set $\boldsymbol{D}_T^{tr}$ of task T into a feature vector z using an encoder g_{ϕ_e} and a correlation network g_{ϕ_r}. The mapping from the feature vector to the base learner parameters $\boldsymbol{w}_T$ of task T is performed using decoder g_{ϕ_d}. The loss function $L_T^{tr}\left(f_{\boldsymbol{w}_T}\right)$ can be calculated for the training set of task T, the loss function of task T can be minimized, and the feature vector $\boldsymbol{z'}_n = z_n - \alpha\nabla_{z_n} L_T^{tr}\left(f_{\boldsymbol{w}_T}\right)$ for each category is updated. The decoder g_{ϕ_d} is used to perform the mapping from the updated feature vector to the updated base learner parameter $\boldsymbol{w'}_T$ for task T. The loss function $L_T^{val}\left(f_{\theta'_T}\right)$ is computed for the validation set of task T. The base learner feeds the updated parameters and loss function values of the validation set into the meta-learner.
4. The meta-parameter $\phi \leftarrow \phi - \eta\nabla_\phi \sum_{\varepsilon} L_T^{val}\left(f_{\theta'_T}\right)$ minimizes the summation of the loss function of the validation set for all tasks T. Input the updated meta-to the base learner and continuously process the novel classification tasks.

IV. RESULTS AND ANALYSES

A. DATA ACQUISITION

The experimental dataset was generated using the software-defined radio (SDR) platform comprising the software development library GNU Radio, which is a free software toolkit for learning, building and deploying software defined radio systems, the hardware platform universal software radio peripheral (USRP), and a PC with a Linux system. Ubuntu 18.04 was used instead of the conventional Linux system. We use USRP B210 devices to collect experimental signal data. The integrated RF front end of the USRP B210 adopts the AD9361, which can transmit real-time RF bandwidth up to 56MHz. In addition, the transmitting power of the USRP B210 is set to 16.5dBm, and both transmitting and receiving antennas are VERT2450 vertical antennas. The SDR platform is shown in Fig. 2.

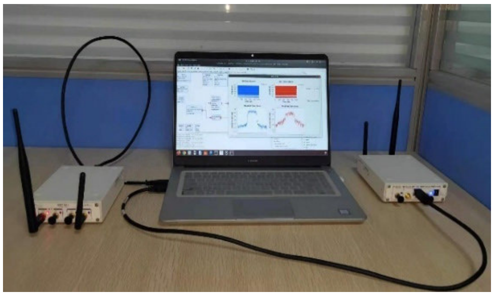


FIGURE 2. Software-defined radio platform.

In order to simplify the experiment, we set up the SDR platform in the office scenario, where the transmitting transmitter and receiving transmitter are 4.5 meters apart, and there are no obstacles between them. Here, seven individual USRP devices of the same model were selected as the specific communication emitters. Six devices were used as transmitting emitters, and one device was used as the receiving emitter connected to two PCs. Signal acquisition and transmission by the communication system are defined by GNU Radio on the PC end, and the RF signal data transmitted by different individual emitters of eight types are finally acquired. The operating frequency of the signal at the transmitter end is set to 2.4 GHz, and the received signal is sampled at the receiver end with a sampling frequency of 16 MHz. Furthermore, the signal modulation method is set to QPSK with a bandwidth of 1.2 MHz.

A frame separation is performed for each type of RF-received sampling signal, with each frame containing 256 sampled data points. Subsequently, each RF signal data frame is preprocessed using MATLAB 2020a to obtain feature vectors as input for network training and testing. The specific program modules of MATLAB 2020a include ’bispecd.m’ and ’Radon.m’, which are respectively used for bispectral analysis and Radon transform. Additionally, data quality is high, considering that the signal transmission and acquisition are performed in a laboratory environment. The identification performance should be tested in environments with different to validate the robustness of the algorithm to noise. Thus, noise should be artificially added to the collected signal data. Here, the acquired raw signal data is fed into MATLAB 2020a to add noise, and the SNR is set to 0 dB, 2 dB,…, 20 dB.

The meta-learning and meta-testing tasks were constructed based on the acquired signal database. For the meta-learning task, a four-way 10-shot task was constructed; that is, four categories were randomly selected from six categories of training RF signals, and 20 training samples (frames) were acquired from each category and labeled. Further, the support and query sets of a task contained 4 × 10 training samples (frames). Consequently, 50 meta-learning tasks were constructed. For the meta-testing task, four categories were randomly selected from six categories of training RF signals, and 50 training samples (frames) were acquired from each category and labeled to form a support set for the fine-tuning of the model parameters. Subsequently, 50 signal samples were acquired from each of the six categories of training RF signals to form a query set to evaluate the performance of the proposed models.

B. IMPACT OF THE NUMBER OF TRAINING SAMPLES ON IDENTIFICATION PERFORMANCE

The number of training samples with labels is an essential factor that influences the identification performance of the algorithm. According to the experimental dataset in Section B, the number of training samples is set to 10, 20, …, 60 for each type of RF signal; subsequently, the identification accuracy of the algorithm is validated, corresponding to different numbers of training samples. The experimental results are shown in Fig. 3.

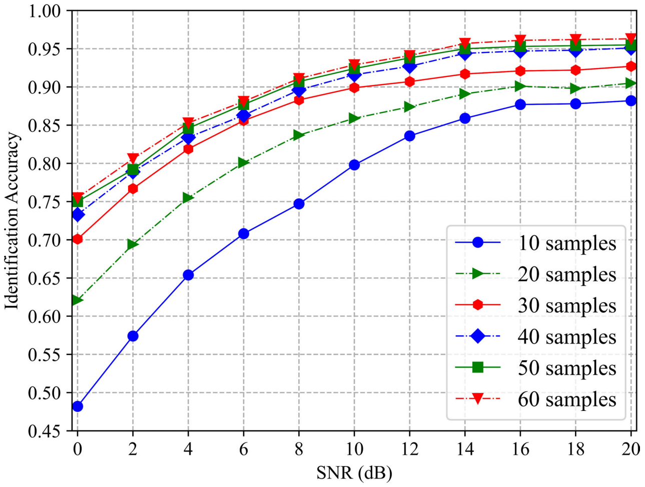


FIGURE 3. Identification accuracy under different numbers of training samples.

As shown in Fig. 3, as the number of training samples increases, the average identification accuracy of the algorithm improves. However, when the number of training samples reaches 40, the overall identification rate of the algorithm is no longer improved and remains relatively stable at a high identification rate. Thus, the algorithm can perform suitable network training with fewer training samples and obtain stable and efficient identification performance that can suitably accommodate the task of SEI in non-cooperative scenarios.

The identification confusion matrix for 50 training samples and SNRs of 0 dB, 8 dB, 16 dB, and 20 dB are shown in Fig. 4(a-d), respectively.

As observed in the confusion matrix, no significant difference exists in the identification accuracy of the designed system for each signal type, and it can achieve a relatively uniform identification of each emitter device. Thus, the proposed method can comprehensively learn the RFFs of each type of signal using fewer training samples. This further demonstrates the effectiveness of the proposed method.

C. IMPACT OF THE MODULATION SCHEME ON IDENTIFICATION PERFORMANCE

The previous experimental validations were conducted using the signal data based on QPSK modulation. Here, the experiments were conducted using the signal data via five different modulation methods: BPSK, QPSK, 16QAM, 8PSK, and 64QAM, and the number of training samples was fixed at 50 to validate the impact of the modulation method on the identification performance of the algorithm. The experimental results are shown in Fig. 5.

As shown in Fig. 5, the overall identification accuracy remains the same when the experiments are conducted using signal data via different modulation methods. Therefore, the modulation methods have a negligible effect on the identification performance of the algorithm, and the algorithm can address the task of SEI under different modulation methods.

D. IMPACT OF THE NUMBER OF THE EMITTERS ON IDENTIFICATION PERFORMANCE

The RF signal data of 6, 8, …, 16 USRP devices were acquired using the SDR platform, and Gaussian white noise with SNRs of 0 dB, 2 dB, …, 20 dB was added. The experimental datasets (that are set up as discussed in Section 4 C) are used to validate the simulation. The identification performance of the proposed algorithm was evaluated using different numbers of communication emitters, and the experimental identification results are shown in Fig. 6.

As shown in Fig. 6, with an increase in the number of communication emitters to be identified, the identification performance of the proposed emitter identification algorithm decreases; however, the decreasing trend is not obvious. Thus, the algorithm can effectively identify a larger number of communication emitter individuals considerably, and the algorithm can be expandable in terms of the identification of the number of emitter individuals. For different SNR ranges, the decrease in the identification accuracy is relatively uniform, and no significant decrease in the identification accuracy was experienced in environments with low SNRs, indicating that the algorithm is robust to noise interference when identifying different numbers of communication emitter individuals.

E. IMPACT OF THE PROPAGATION CHANNEL ON IDENTIFICATION PERFORMANCE

According to the experimental dataset (established as discussed in Section 4 C), the AWGN, Rayleigh, and Rice noise are added to simulate the RF signal passing through the AWGN, Rayleigh, and Rice channel. The identification performance of the proposed communication emitter individual identification algorithm under different communication transmission channels was tested. The experimental identification results are shown in Fig. 7.

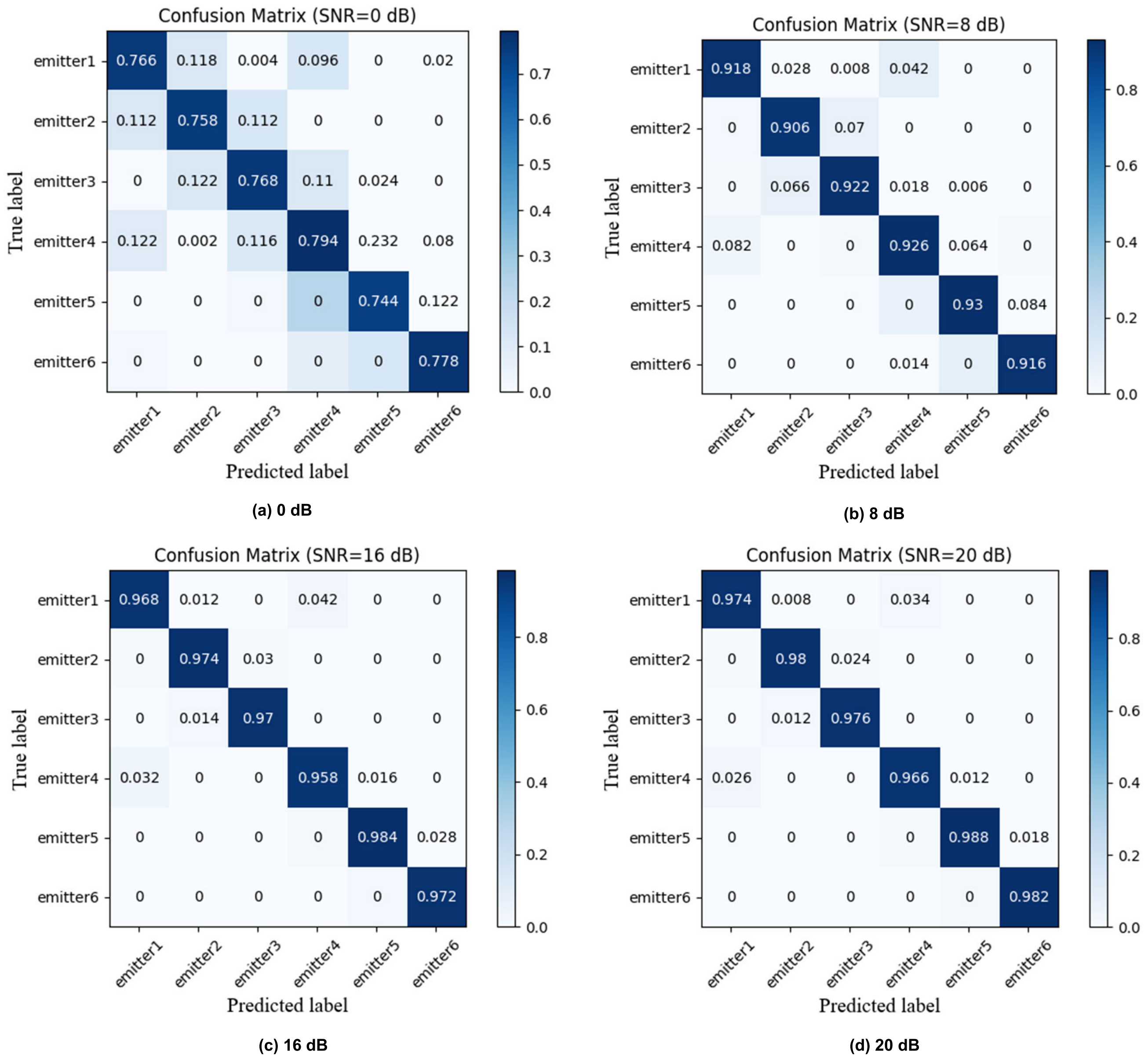
As shown in Fig. 7, the identification performance of the algorithm deteriorates more after adding Rayleigh and Rice noise to the RF signal data than in the addition of AWGN noise, particularly in an environment of low SNR. As the SNR increases, the difference between the identification accuracy of the algorithm under two types of noise interference is reduced. Furthermore, the identification accuracy of the algorithm reaches 0.9 at 10 dB SNR in the Rayleigh channel and 0.9 at 12 dB SNR in the Rice channel, indicating that the algorithm is not significantly influenced by the Rayleigh and Rice channels. Thus, it can better adapt to the task of individual identification of the communication emitters in various transmission channels.

F. OTHER INDICES FOR THE EVALUATION OF THE IDENTIFICATION PERFORMANCE

Identification accuracy is the ratio of the number of correctly identified samples to the number of total samples tested; it only reflects the overall identification performance. However, determining whether each category of RF signals is correctly identified is difficult, particularly when each category constitutes the minority relative to the other RF signals, leading to the category imbalance problem. Here, the identification accuracy is not a comprehensive evaluation index of the identification performance.

In this section, two methods, including receiver operating characteristic (ROC) and precision-recall, are used to further evaluate the identification performance of the proposed method. For the six USRP devices to be identified, one device is chosen as a positive class with a weight of 5. The remaining five devices as a single negative class with a weight of 1. Six types of RF signals with an SNR of 10 dB are acquired for training and identification. The ROC and precision-recall curves for the six devices are shown in Fig. 8 and 9, respectively. The area under the curve (AUC) of each device was also calculated to visualize their correct identification rate.

As shown in Figs. 8 and 9, the ROC and precision-recall AUCs of each device are greater than 90%, further indicating that the algorithm exhibits suitable identification performance.



(a) 0 dB

(b) 8 dB

(c) 16 dB

(d) 20 dB

FIGURE 4. Identification confusion matrix when the number of training samples is 50, and the signal-to-noise ratio is 0 dB, 8 dB, 16 dB, and 20 dB, respectively.

G. COMPARISON WITH OTHER IDENTIFICATION METHODS

In this section, we demonstrate the superiority of the proposed method, and the proposed method was compared with the conventional deep learning model, the previously proposed MSRI [26], and meta-learning models [27] to test their identification performance. The number of training samples is fixed at 50, the modulation method is set to QPSK, the number of emitters is fixed at 6, and the propagation channel is set to AWGN.

The experimental results are shown in Fig. 10.

As shown in Fig. 10, the identification accuracy of the proposed algorithm is greater than 80% at an SNR of 2 dB and greater than 90% at an SNR of 8 dB, indicating that the algorithm can achieve high accuracy of open set identification in an environment with low or medium SNR. In contrast, the MSRI method requires a backward shift of the SNR of 2 dB to achieve the same identification accuracy; the meta-learning model also requires a backward shift of SNR of 4 dB to achieve the same identification accuracy. Moreover, the identification performance of the conventional deep learning model deteriorated severely under the condition of few-shot, and the identification accuracy could only reach over 80% at an SNR of 16 dB. Therefore, the proposed method outperforms the existing few-shot SEI method, validating the superiority of the proposed method in few-shot SEI.

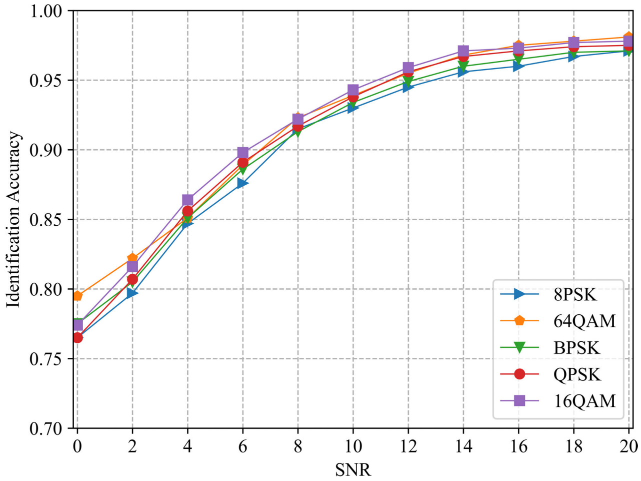


FIGURE 5. Identification accuracy of different modulation methods.

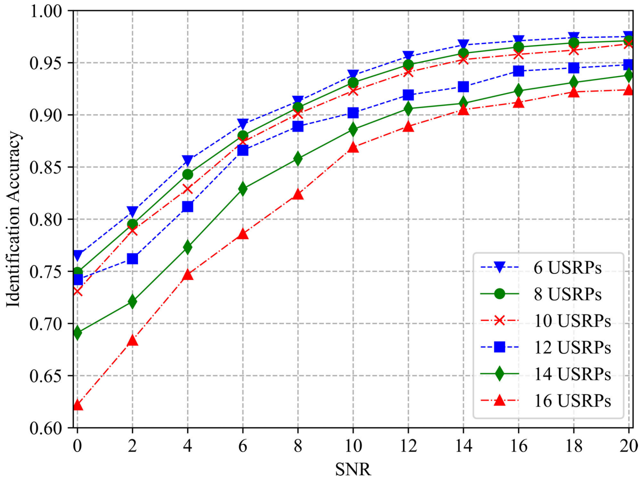


FIGURE 6. Identification accuracy with different numbers of communication emitter individuals.

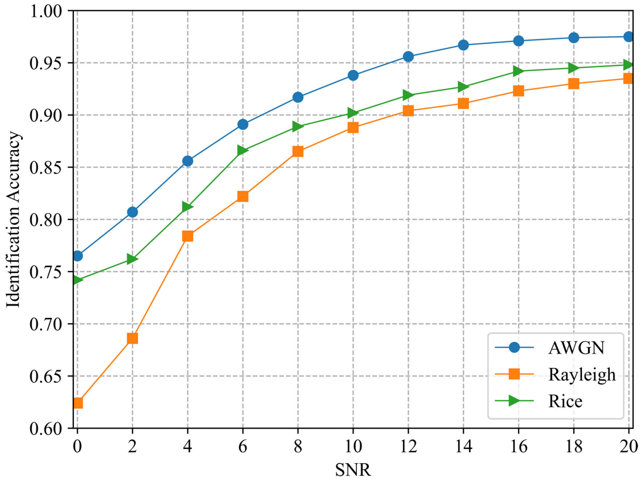


FIGURE 7. Correct identification rate under different transmission channels.

Furthermore, six types of RF signals were acquired, the SNR was set to 10 dB, the number of training samples for each type of signal was set to 50, and simulation experiments were conducted for validation. The ROC curves corresponding to the three methods are plotted, as shown in Fig. 11.

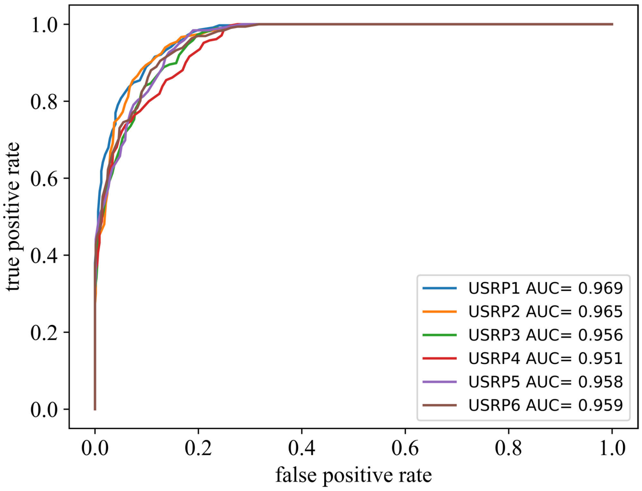


FIGURE 8. ROC curves for each of the six devices.

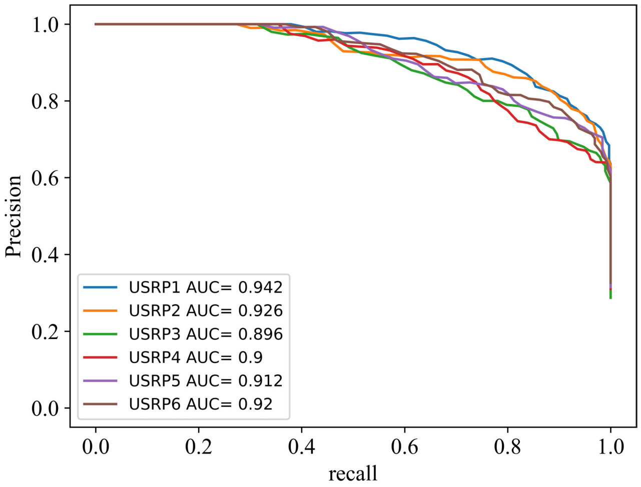


FIGURE 9. Precision-recall curves for each of the six devices.

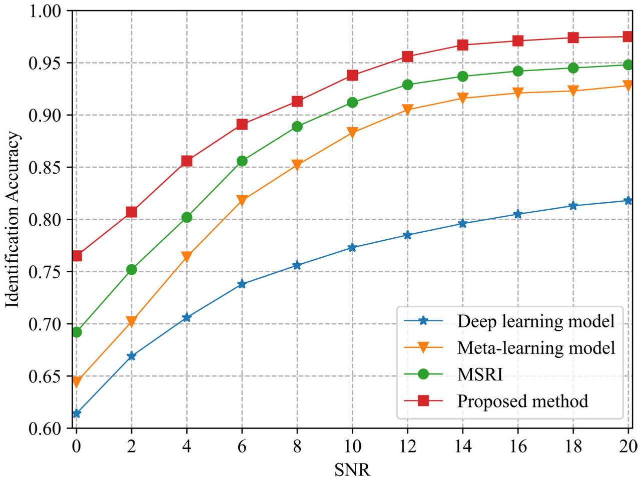
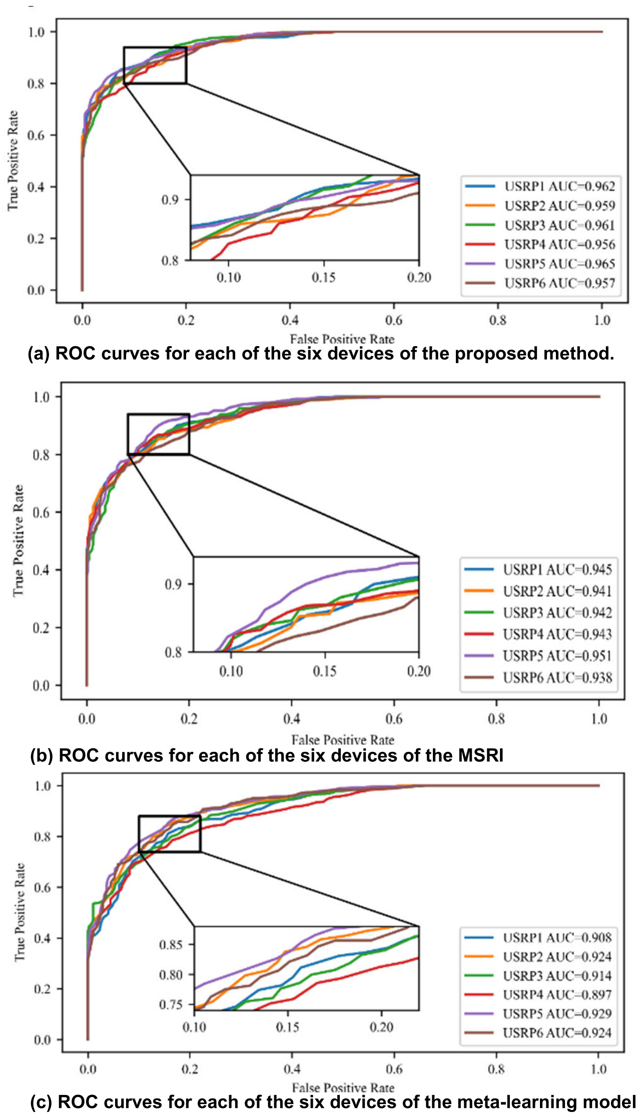


FIGURE 10. Comparison of the identification performance of the proposed method via other methods.



(a) ROC curves for each of the six devices of the proposed method.

(b) ROC curves for each of the six devices of the MSRI

(c) ROC curves for each of the six devices of the meta-learning model

FIGURE 11. ROC curves for each of the six devices of the proposed method, MSRI, and meta-learning model.

As shown in Fig. 11, the ROC curves of all three methods are distributed in the upper left region of the figure, and the ROC curves of the proposed method are more concentrated in the upper left region of the figure compared with the other two methods, indicating that the proposed method exhibits a higher true positive rate and a lower fake positive rate in identifying each emitter. Additionally, the proposed method achieves the maximum ROC AUC, further demonstrating its superior identification performance.

V. DISCUSSION AND CONCLUSIONS

We proposed a novel few-shot SEI method and developed a bispectral analysis and Randon transformation-based signal preprocessing scheme to highlight the RFFs of the received signal. Further, we applied a meta-learning model under few-shot conditions to address the SEI conundrum. Moreover, an LEO algorithm-based enhanced meta-learning model was proposed. The LEO algorithm extracted low-dimensional key features from the high-dimensional input data and calculated the distance and degree of dispersion between these features. This information was applied in the sample point prediction. The LEO algorithm considered the sample point similarities and dispersion degrees, integrated the sample point distribution features, and normalized the distances. Consequently, comparable sample point distances were observed between different categories and accurate classification. The proposed method exhibited stable and efficient recognition upon training with a minimum of forty samples and identified the individual communication radiation sources under multiple modulation types. Moreover, it exhibited scalability in identifying the emitter numbers. Finally, this method exhibited adaptability in identifying the emitter individuals under multiple propagation channel types.

In future work, we will study whether the algorithm provided in this manuscript is suitable for devices configured with multiple antennas. In addition, real-world RF signals captured for experimental verifications to justify further and improve the versatility of the proposed method.

REFERENCES

[1] X. Wang, Y. Zhang, H. Zhang, Y. Li, and X. Wei, "Radio frequency signal identification using transfer learning based on LSTM," *Circuits, Syst., Signal Process.*, vol. 39, no. 11, pp. 5514–5528, Nov. 2020, doi: 10.1007/s00034-020-01417-7.

[2] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018, doi: 10.1109/JSTSP.2018.2796446.

[3] X. Tian, X. Wu, H. Li, and X. Wang, "RF fingerprints prediction for cellular network positioning: A subspace identification approach," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 450–465, Feb. 2020, doi: 10.1109/TMC.2019.2893278.

[4] S. Rajendran, Z. Sun, F. Lin, and K. Ren, "Injecting reliable radio frequency fingerprints using metasurface for the Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1896–1911, 2007, doi: 10.1109/TIFS.2020.3045318.

[5] M. O. Moore, R. M. Buehrer, and W. C. Headley, "Decoupling RNN training and testing observation intervals for spectrum sensing applications," *Sensors*, vol. 22, no. 13, p. 4706, Jun. 2022, doi: 10.3390/s22134706.

[6] W. Jiang, W. Yu, W. Wang, and T. Huang, "Multi-agent reinforcement learning for joint cooperative spectrum sensing and channel access in cognitive UAV networks," *Sensors*, vol. 22, no. 4, p. 1651, Feb. 2022, doi: 10.3390/s22041651.

[7] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Sep. 2005, doi: 10.1109/JSAC.2004.839380.

[8] P. Padilla, J. L. Padilla, and J. F. Valenzuela-Valdés, "Radiofrequency identification of wireless devices based on RF fingerprinting," *Electron. Lett.*, vol. 49, no. 22, pp. 1409–1410, 2013, doi: 10.1049/el.2013.2759.

[9] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *J. Commun. Netw.*, vol. 11, no. 6, pp. 544–555, Dec. 2009, doi: 10.1109/JCN.2009.6388408.

[10] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet denoising to improve OFDM-based signal detection and classification," *Secur. Commun. Netw.*, vol. 3, no. 1, pp. 71–82, Jan. 2010, doi: 10.1002/sec.115.

[11] Y. Yuan, Z. Huang, H. Wu, and X. Wang, "Specific emitter identification based on Hilbert–Huang transform-based time-frequency-energy distribution features," *IET Commun.*, vol. 8, no. 13, pp. 2404–2412, Sep. 2014, doi: 10.1049/iet-com.2013.0865.

[12] J. Zhang, F. Wang, O. A. Dobre, and Z. Zhong, "Specific emitter identification via Hilbert–Huang transform in single-hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1192–1205, Jun. 2016, doi: 10.1109/TIFS.2016.2520908.

[13] U. Satija, N. Trivedi, G. Biswal, and B. Ramkumar, "Specific emitter identification based on variational mode decomposition and spectral features in single hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 581–591, Mar. 2019, doi: 10.1109/TIFS.2018.2855665.

[14] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017, doi: 10.1109/TCCN.2017.2758370.

[15] L. Han, J. Sun, and W. Zhang, "Convolutional neural network for convective storm nowcasting using 3-D Doppler weather radar data," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1487–1495, Feb. 2020, doi: 10.1109/TGRS.2019.2948070.

[16] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.

[17] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18484–18501, 2018, doi: 10.1109/ACCESS.2018.2818794.

[18] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018, doi: 10.1109/COMST.2018.2846401.

[19] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2591–2594, Dec. 2018, doi: 10.1109/LCOMM.2018.2871465.

[20] Y. Xu, G. Feng, and Y. Zhao, "One improvement to two-dimensional locality preserving projection method for use with face recognition," *Neurocomputing*, vol. 73, nos. 1–3, pp. 245–249, Dec. 2009, doi: 10.1016/j.neucom.2009.09.010.

[21] Y. Pan, S. Yang, H. Peng, T. Li, and W. Wang, "Specific emitter identification based on deep residual networks," *IEEE Access*, vol. 7, pp. 54425–54434, 2019, doi: 10.1109/ACCESS.2019.2913759.

[22] L. J. Wong, W. C. Headley, and A. J. Michaels, "Specific emitter identification using convolutional neural network-based IQ imbalance estimators," *IEEE Access*, vol. 7, pp. 33544–33555, 2019, doi: 10.1109/ACCESS.2019.2903444.

[23] B. X. He and F. G. Wang, "Cooperative specific emitter identification via multiple distorted receivers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3791–3806, 2020, doi: 10.1109/TIFS.2020.3001721.

[24] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O. A. Dobre, and H. V. Poor, "An efficient specific emitter identification method based on complex-valued neural networks and network compression," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2305–2317, Aug. 2021, doi: 10.1109/JSAC.2021.3087243.

[25] X. Zha, H. Chen, T. Li, Z. Qiu, and Y. Feng, "Specific emitter identification based on complex Fourier neural network," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 592–596, Mar. 2022, doi: 10.1109/LCOMM.2021.3135378.

[26] Y. Qian J. Qi, X. Kuai, G. Han H. Sun, and S. Hong "Specific emitter identification based on multi-level sparse representation in automatic identification system," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2872–2884, 2021, doi: 10.1109/TIFS.2021.3068010.

[27] N. Yang, B. Zhang, G. Ding, Y. Wei, G. Wei, J. Wang, and D. Guo, "Specific emitter identification with limited samples: A model-agnostic meta-learning approach," *IEEE Commun. Lett.*, vol. 26, no. 2, pp. 345–349, Feb. 2022, doi: 10.1109/LCOMM.2021.3110775.

[28] Z. Sun, S. Kong, and W. Wang, "Sparse learning of higher-order statistics for communications and sensing," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 1, pp. 13–22, Feb. 2020, doi: 10.1109/TETCI.2018.2872404.

[29] L. Zhang, Y. Wang, and D. Zhang, "Research on multiple-image encryption mechanism based on radon transform and ghost imaging," *Opt. Commun.*, vol. 504, Feb. 2022, Art. no. 127494, doi: 10.1016/j.optcom.2021.127494.

[30] A. Gholami and M. Farshad, "Fast hyperbolic Radon transform using chirp-Z transform," *Digit. Signal Process.*, vol. 87, pp. 34–42, Apr. 2019, doi: 10.1016/j.dsp.2019.01.003.

[31] T. Mu, H. Wang, C. Wang, Z. Liang, and X. Shao, "Auto-CASH: A meta-learning embedding approach for autonomous classification algorithm selection," *Inf. Sci.*, vol. 591, pp. 344–364, Apr. 2022, doi: 10.1016/j.ins.2022.01.040.

CUNXIANG XIE was born in Yantai, Shandong, China, in 1996. He received the B.S. degree in communication engineering from Naval Aviation University, in 2019, where he is currently pursuing the M.S. degree in information and communication engineering with the Department of Information Fusion. His research interests include deep learning and specific emitter identification.

LIMIN ZHANG was born in Kaiyuan, Liaoning, China, in 1966. He received the Ph.D. degree in signal processing technology from Tianjin University, in 2005. Since 2005, he has been a Professor with Naval Aviation University. His research interest includes satellite communication signal processing.

ZHAOGEN ZHONG was born in Nanchang, Jiangxi, China, in 1984. He received the Ph.D. degree in information and communication engineering from Naval Aviation University, in 2013. He is currently an Associate Professor with Naval Aviation University. His research interest includes spread spectrum signal processing.